

## **IN THE CLAIMS:**

The following listing of claims will replace all prior listings of claims in the application:

1. (Previously Presented): A method of communicating between a TCP stack, wherein the TCP stack delegates one or more connections to the offload unit, and the TCP stack processes connections that are not delegated or require special processing, and an offload unit, comprising:
  - utilizing a driver as a translator for writing a command including an index corresponding to a delegated connection to an entry in a command ring;
  - indicating by a bit an owner of the entry, the owner being the offload unit, wherein every entry in the ring includes a bit indicating the owner of the entry;
  - reading the command from the entry in the command ring to the offload unit;
  - setting the bit in the entry by the offload unit to indicate the owner of entry is the TCP stack;
  - executing the command; and
  - writing command specific status to the entry in the command ring by the offload unit.
2. (Original): The method of claim 1, wherein the command includes a location of a buffer for storing payload data produced by the offload unit.
3. (Original): The method of claim 1, wherein the command includes connection information needed to setup a delegated connection.
4. (Original): The method of claim 1, wherein the command specific status includes a value representing a number of buffers accepted by the offload unit.
5. (Original): The method of claim 1, further comprising:
  - writing a notification descriptor including an index corresponding to a delegated connection to an entry in a notification ring; and

reading the notification descriptor from the entry in the notification ring.

6. (Original): The method of claim 5, wherein the notification descriptor includes one or more notification flags indicating specific information for a connection.

7. (Currently Amended): A method of communicating between a TCP stack and an offload unit, wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing comprising:

writing a notification descriptor to an entry in a notification ring by the offload unit;  
indicating by a bit an owner of the entry, the owner-being the TCP stack, wherein every entry in the ring includes a bit indicating the owner of the entry;  
utilizing a driver as a translator for reading the notification descriptor from the entry in the notification ring;  
reading the notification descriptor from the entry in the notification ring to the ~~the~~ TCP stack based on the bit; and  
determining connection information for a delegated connection based on the notification descriptor.

8. (Original): The method of claim 7, wherein the notification descriptor includes an index corresponding to the delegated connection.

9. (Currently Amended): The method of claim 7, wherein the notification descriptor includes one or more notification flags corresponding to the delegated connection.

10. (Original): The method of claim 7, wherein the notification descriptor includes a count of received acknowledgements.

11. (Original): The method of claim 8, wherein a flag indicates an acknowledgement threshold was reached on the delegated connection.

12. (Original): The method of claim 8, wherein a flag indicates a duplicate acknowledgement was received on the delegated connection.
13. (Original): The method of claim 8, wherein a flag indicates a sequence number threshold was reached on the delegated connection.
14. (Original): The method of claim 8, wherein a flag indicates at least a portion of frame data received on the delegated connection was uploaded by the offload unit to a legacy buffer.
15. (Original): The method of claim 8, wherein a flag indicates a request for a user buffer for uploading of payload data from the offload unit.
16. (Previously Presented): A system for transmitting commands from a TCP stack wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing, to an offload unit, comprising:
- a command ring configured to receive commands written by the TCP stack;
  - a driver configured as a translator for writing a command including an index corresponding to a delegated connection to an entry in ~~[[a]]the~~ command ring;
  - the driver indicating by a bit, the offload unit as an owner of the entry that is written with the command, wherein every entry in the command ring includes a bit indicating the owner of the entry; and
  - the offload unit configured to read entries from the command ring that indicate the offload unit as the owner and to process the commands for the one or more connections delegated to the offload unit and set the bit indicating that the TCP stack is the owner in any entries that are read by the offload unit.
17. (Original): The system of claim 16, further comprising:

a notification ring configured to receive connection information written by the offload unit and output connection information read by the TCP stack.

18. (Original): The system of claim 16, wherein the offload unit is configured to write command specific status to the command ring.

19. (Original): The system of claim 16, further comprising a transmit descriptor ring configured to transfer transmit buffer information from the TCP stack to the offload unit.

20. (Original): The system of claim 19, wherein the transmit buffer information includes a delegated connection index.

21. (Original): The system of claim 16, further comprising a receive descriptor ring configured to transfer receive buffer information from the TCP stack to the offload unit.

22. (Currently Amended): A system for transmitting connection information from an offload unit to a TCP stack wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing, comprising:

a notification unit within the offload unit configured to write notification descriptors to entries in a notification ring;

the offload unit indicating by a bit an owner of each entry written by the notification unit as the TCP stack, wherein every entry in the notification ring includes a bit indicating the owner of the entry;

a driver configured as a translator for writing a command including an index corresponding to a delegated connection to an entry in a command ring;

the TCP stack indicating by a bit an owner of each entry of the command ring that is written by the TCP stack as the offload unit, wherein every entry in the command ring includes a bit indicating the owner of the entry;

the notification ring configured to output notification descriptors read by the TCP stack from the entries that indicate the TCP stack as the owner; and

the TCP stack configured to process the notification descriptors for the one or more connections that are delegated to the offload unit and set the bit indicating that the offload unit is the owner in any entries in the notification ring that are read by the TCP stack.

23. (Original): The system of claim 22, wherein the notification unit is configured to set one or more flags, each flag indicating connection information.

24. (Previously Presented): The system of claim 22, wherein the notification unit is configured to set a flag when at least a portion of a partially parsed frame is uploaded to a legacy buffer.

25. (Previously Presented): The system of claim 22, wherein each notification descriptor includes a sequence number.

26. (Previously Presented): The system of claim 22, wherein each notification descriptor includes an acknowledgement number.

27. (Previously Presented): The system of claim 22, wherein each notification descriptor includes a count of received acknowledgements.

28-31. (Cancelled)

32. (Currently Amended): ~~[[A]]The method of as claimed in claim 1, further comprising:~~  
writing wherein when the TCP stack writes the entry to the command ring[,] by the TCP stack; and  
setting the bit in the entry pointer comprising Command Write Pointer is set to indicate that the entry is owned by the offload unit.

33. (Currently Amended): ~~[[A]]The method of as claimed in~~ claim 1, wherein when the offload unit writes the entry to the command ~~ring unit~~, the ~~bit in the entry pointer~~ comprising command read pointer is set to indicate that the entry is owned by the TCP stack.

34. (Currently Amended): ~~[[A]]The method of as claimed in~~ claim 33, wherein when the offload unit writes the entry to the command unit, the ~~bit in the entry pointer~~ comprising Command Read Pointer is set to indicate that the entry is owned by the TCP stack.

35. (Currently Amended): ~~[[A]]The method of as claim in~~ claim 31, wherein at startup the ~~bit pointer~~ indicates the entries owned by the TCP stack.

36. (Cancelled)

37. (Currently Amended): ~~[[A]]The method of as claims in~~ claim 36, including the step of the offload unit using the ring to notify the TCP stack of offloaded connections needing further processing by the TCP stack.

38. (Previously Presented): The system of claim 22, wherein the offload unit is configured to process the commands for the one or more connections that are delegated to the offload unit and set the bit indicating that the TCP stack unit is the owner in any entries in the command ring that are read by the offload unit.

39. (Previously Presented): The method of claim 2, wherein the command includes a synchronization bit configured to enable the offload unit to accept user buffer descriptors that specify locations of buffers for storing payload data produced by the offload unit.

40. (Previously Presented): The method of claim 5, further comprising setting a bit in the entry in the notification ring by the offload unit indicating that the entry is owned by the TCP stack when the notification descriptor is written.

41. (Previously Presented): The method of claim 5, further comprising:  
    setting a synchronization request flag in the notification descriptor; and  
    flushing unused user buffer descriptors queued in the offload unit for the delegated connection, wherein the unused buffer descriptors specify locations of buffers for storing payload data produced by the offload unit.